<u>CLAIMS</u>

1. A method implementing a robust 2-phase commit protocol between a client and a server via a relational table and software facilitating communications with said client and said server, said relational table storing a list of potentially indoubt units of work, said

5   method as implemented in said software comprising the steps of:

(a) receiving an invocation from said client for a first phase of commit for a transaction representing a unit of work;

(b) inserting an entry in said relational table corresponding to said unit of work and transmitting an instruction to said server to prepare to commit for said transaction,

10   said inserted entry indicating said unit of work is potentially an indoubt entry; and

(c) receiving a request from said client to perform any of the following decisions: a COMMIT, a ROLLBACK, or a RECOVER, wherein said relational table is updated after execution of said request.

15   2. A method as per claim 1, wherein, if said received request is a COMMIT or a ROLLBACK decision, said method comprising the steps of:

communicating with said server and processing said COMMIT or ROLLBACK request, and upon successful processing,

deleting a corresponding entry in said relational table.

20

3. A method as per claim 1, wherein, if said received request is a RECOVER decision, said method comprising the steps of:

    querying said relational table to identify a list of indoubt units of work;

    transmitting said list of indoubt units of work to said client;

5   receiving a COMMIT or ROLLBACK decision from said client;

    communicating with said server to process said commit or rollback request, and upon successful processing,

    deleting a corresponding entry in said relational table.

10 4. A method as claim 1, wherein said server is a database cluster and said software supports execution of said recover decision even if one or more members of the database cluster are unavailable.

5. A method as per claim 1, wherein said relational table specifies row-level locking for

15 keeping contention on said relation table to a minimum.

6. A method as per claim 1, wherein said relational table is a SQL table and said step of inserting an entry in said relational table is performed via issuing a SQL INSERT instruction.

20

7. A method as per claim 1, wherein said relational table is stored in said server and a request for said insertion of entry in said relational table is placed on a network message that includes said instruction to said server to prepare to commit for said transaction.

8. A method as per claim 1, wherein said method is implemented across networks.

9. A method as per claim 8, wherein said across networks element comprises any of, or a combination of, the following: local area network (LAN), wide area network, wireless network, or the Internet.

10. A method as per claim 1, wherein requests for deletions of entries in said relational table are placed on a separate network connection to avoid starting a new unit of work.

11. A method as per claim 1, wherein said method comprises the step of mapping said 2-phase protocol onto a protocol supported by said server.

12. An article of manufacture comprising computer usable medium having computer readable program code embodied therein implementing a robust 2-phase commit protocol between a client and a server via a relational table and software facilitating communications with said client and said server, said relational table storing a list of potentially indoubt units of work, said medium comprising:

(a) computer readable program code receiving an invocation from said client for a first phase of commit for a transaction representing an unit of work;

(b) computer readable program code inserting an entry in said relational table corresponding to said unit of work and transmitting an instruction to said server to prepare to commit for said transaction, said inserted entry indicating said unit of work is potentially an indoubt entry; and

(c) computer readable program code aiding in receiving a request from said client to perform any of the following decisions: a COMMIT, a ROLLBACK, or a RECOVER, wherein said relational table is updated after execution of said request.

13. An article of manufacture as per claim 12, said medium further comprising:

computer readable program code aiding in communicating with said server and processing said COMMIT or ROLLBACK request, and upon successful processing,

computer readable program code deleting a corresponding entry in said relational table.

14. An article of manufacture as per claim 12, said medium further comprising:

computer readable program code querying said relational table to identify a list of indoubt units of work;

computer readable program code aiding in transmitting said list of indoubt units of work to said client;

computer readable program code aiding in receiving a COMMIT or ROLLBACK decision;

computer readable program code aiding in communicating with said server to process said COMMIT or ROLLBACK request, and upon successful processing, and

5   computer readable program code deleting a corresponding entry in said relational table.

15. An article of manufacture as per claim 12, wherein said relational table specifies row-level locking for keeping contention on said relation table to a minimum.

10

16. An article of manufacture as per claim 12, wherein said relational table is a SQL table and computer readable program code inserts an entry in said relational table is performed via issuing a SQL INSERT instruction.

15   17. An article of manufacture as per claim 12, wherein said relational table is stored in said server and a request for said insertion of entry in said relational table is placed on a network message that includes said instruction to said server to prepare to commit for said transaction.

20   18. A method implementing a robust 2-phase commit protocol between a transaction manager and a database cluster via software facilitating communications with said

transaction manager and said database cluster, said method as implemented in said software comprising the steps of:

(a) creating an SQL table for storing a list of potentially indoubt units of work

(b) receiving an invocation from said transaction manager for a first phase of

5 commit for a transaction representing an unit of work;

(b) inserting, via an SQL INSERT instruction, an indoubt entry in said SQL table corresponding to said unit of work and transmitting an prepare to commit instruction to said database cluster, said SQL INSERT instruction and said prepare to commit instruction placed on one network message to minimize cost; and

10 (c) receiving a request from said transaction manager to perform any of the following decisions: a COMMIT, a ROLLBACK, or a RECOVER, wherein said SQL table is updated after execution of said request.

19. A method as per claim 18, wherein if said received request is a COMMIT or

15 ROLLBACK decision, said method comprising the additional steps of:

communicating with said database cluster and processing said COMMIT or ROLLBACK request, and upon successful processing,

deleting a corresponding entry in said relational table via an SQL DELETE instruction.

20

20. A method as per claim 19, wherein said SQL DELETE instructions are paced on a
separate network connection to avoid starting a new unit of work.

21. A method as per claim 18, wherein if said received request is a RECOVER decision,

5   said method comprising the additional steps of:

querying said SQL table to identify a list of indoubt units of work;

transmitting said list of indoubt units of work to said transaction manager;

receiving a commit or rollback decision from said transaction manager;

communicating with said database cluster to process said commit or rollback

10   request, and upon successful processing, and

deleting a corresponding entry in said SQL table via a SQL DELETE instruction.

22. A method as per claim 21, wherein said SQL DELETE instructions are paced on a
separate network connection to avoid starting a new unit of work.

15

23. A method as per claim 18, wherein said SQL table specifies row-level locking for
keeping contention on said relation table to a minimum.

24. A method as per claim 18, wherein said method is implemented across networks.

20

25. A method as per claim 24, wherein said across networks element comprises any of, or a combination of, the following: local area network (LAN), wide area network, wireless network, or the Internet.

5 26. Software implementing a 2-phase commit protocol between a client and a server comprising:

a first module invoked to create a relational table in said server to store potential indoubt units of work;

a second module invoked to insert or delete indoubt entries of work in said

10 relational table,

wherein insertions of indoubt entries are performed if an invocation is received from said client for a first phase of commit for a transaction representing a unit of work; and

wherein deletions of indoubt entries are performed upon successful

15 processing of a commit or rollback decision; and

a third module invoked upon receiving a recover instruction from said client, said third module extracting a list of indoubt units of work from said relational table and transmitting said extracted list to said client, wherein said client inspects said list and issues a commit or rollback decision to said middlware regarding said indoubt units of

20 work in said list.

27. Software as claim 26, wherein said server is a database cluster and said software supports execution of said recover decision even if one or more members of the database cluster are unavailable.

5   28. Software as per claim 26, wherein said relational table specifies row-level locking for keeping contention on said relation table to a minimum.

29. Software as per claim 26, wherein said relational table is a SQL table and said step of inserting an entry in said relational table is performed via issuing a SQL INSERT
10   instruction.

30. Software as per claim 26, wherein a request for said insertion of entry in said relational table is placed on a network message that includes an instruction to said server to prepare to commit for said transaction.

15

31. Software as per claim 26, wherein said software further comprises a fourth module mapping said 2-phase protocol onto a protocol supported by said server.